



Enter your query below, and then click *Start Search*

(synchronizing or timing or time adjustment)

Collections to Search:

☒ Journals ☒ Conference Proceedings ☒ Standards

Additional Criteria:

☐ Match similar subject terms ☐ Latest additions to IEL

Publication Year(s): to

Results List: Sort by in order

List results at a time Display line citation

If you have any question about IEL Online and how it works, check the Frequently Asked Questions or email [IEL Online Support](#).

To let us know what you think about IEL Online, email your comments to [IEEE Electronic Products](#)

[| IEL Online Home](#) | [Search](#) | [Advanced Search](#) | [What's New](#) | [Help](#) | [Logout](#) |
[| FAQ's](#) | [Support](#) | [Comments](#) |













Copyright 1999 Institute of Electrical and Electronics Engineers. All rights reserved.









































Collection: ☒ Journals ☒ Conferences ☒ Standards

Your search matched **11795** of **608497** documents.

25 are presented on this page, sorted by Score in descending order.

DOC TYPE	VIEW ISSUE TOC	VIEW FULL PAGE	VIEW CITATION
PER			<p><u>An accurate worst case timing analysis for RISC processors</u> <i>Sung-Soo Lim; Young Hyun Bae; Gyu Tae Jang; Byung-Do Rhee; Sang Lyul Min; Chang Yun Park; Heonshik Shin; Kunsoo Park; Soo-Mook Moon; Chong Sang Kim</i> Software Engineering, IEEE Transactions on Volume: 21 7 , July 1995 , Page(s): 593 -604</p>
CNF			<p><u>Synchronization in high-speed telecommunications networks</u> <i>Howe, C.F.</i> Computers and Communications, 1995., Conference Proceedings of the 1995 IEEE Fourteenth Annual International Phoenix Conference on , 1995 , Page(s): 566 -570</p>
CNF			<p><u>Hierarchical timing-driven floorplanning and place and route using a timing budgeter</u> <i>Venkatesh, S.V.</i> Custom Integrated Circuits Conference, 1995., Proceedings of the IEEE 1995 , 1995 , Page(s): 469 -472</p>
CNF			<p><u>An accurate worst case timing analysis technique for RISC processors</u> <i>Sung-Soo Lim; Young Hyun Bae; Gyu Tae Jang; Byung-Do Rhee; Sang Lyul Min; Chang Yun Park; Heonshik Shin; Kunsoo Park; Chong Sang Kim</i> Real-Time Systems Symposium, 1994., Proceedings. , 1994 , Page(s): 97 -108</p>
CNF			<p><u>A modular timing system for megawatt gyrotrons</u> <i>Petersen, D.E.</i> Fusion Engineering, 1991. Proceedings., 14th IEEE/NPSS Symposium on , 1992 , Page(s): 835 -838 vol.2</p>
PER			<p><u>Flow synchronization protocol</u> <i>Escobar, J.; Partridge, C.; Deutsch, D.</i> Networking, IEEE/ACM Transactions on Volume: 2 2 , April 1994 , Page(s): 111 -121</p>

- | | | | |
|-----|---|---|---|
| CNF |  |  | <p><u>Static timing analysis for self resetting circuits</u>
 <i>Narayanan, V.; Chappell, B.A.; Fleischer, B.M.</i>
 Computer-Aided Design, 1996. ICCAD-96. Digest of Technical Papers., 1996 IEEE/ACM International Conference on , 1996 , Page(s): 119 -126</p> |
| PER |  |  | <p><u>Using timed CSP for specification verification and simulation of multimedia synchronization</u>
 <i>Ates, A.F.; Bilgic, M.; Saito, S.; Sarikaya, B.</i>
 Selected Areas in Communications, IEEE Journal on Volume: 14 1 , Jan. 1996 , Page(s): 126 -137</p> |
| CNF |  |  | <p><u>Modeling data transfer signal timings in DAME</u>
 <i>Huber, B.; Li, K.; Dimopoulos, N.; Escalante, M.; Manning, E.</i>
 Communications, Computers and Signal Processing, 1993., IEEE Pacific Rim Conference on Volume: 2 , 1993 , Page(s): 505 -509 vol.2</p> |
| CNF |  |  | <p><u>Timing jitter measurement method for a kHz regenerative amplifier system</u>
 <i>Miura, T.; Kobayashi, K.; Zhigang Zhang; Torizuka, K.; Kannari, F.</i>
 Lasers and Electro-Optics, 1999. CLEO/Pacific Rim '99. The Pacific Rim Conference on Volume: 2 , 1999 , Page(s): 413 -414 vol.2</p> |
| CNF |  |  | <p><u>Media synchronization protocols for packet audio-video system on multimedia information networks</u>
 <i>Shibata, Y.; Seta, N.; Shimizu, S.</i>
 System Sciences, 1995. Vol. III,. Proceedings of the Twenty-Eighth Hawaii International Conference on , 1995 , Page(s): 594 -601 vol.2</p> |
| PER |  |  | <p><u>An EFSM-based multimedia synchronization model and the authoring system</u>
 <i>Chung-Ming Huang; Chung-Ming Lo</i>
 Selected Areas in Communications, IEEE Journal on Volume: 14 1 , Jan. 1996 , Page(s): 138 -152</p> |
| CNF |  |  | <p><u>Joint linear timing and carrier phase estimation of DS-CDMA multiuser communications</u>
 <i>Cha'o-Ming Chang; Kwang-Cheng Chen</i>
 Communications, 1999. ICC '99. 1999 IEEE International Conference on , 1999 , Page(s): 1521 -1525 vol.3</p> |
| CNF |  |  | <p><u>Synchronization paradigm for protocol testing under multi-party configuration</u>
 <i>Lin, R.S.; Yuang, M.C.</i>
 Global Telecommunications Conference, 1998. GLOBECOM 1998. The Bridge to Global Integration. IEEE Volume: 6 , 1998 , Page(s): 3460 -3466 vol.6</p> |
| CNF |  |  | <p><u>A statistical method for time synchronization of computer clocks with precisely frequency-synchronized oscillators</u>
 <i>Yamashita, T.; Ono, S.</i>
 Distributed Computing Systems, 1998. Proceedings. 18th International Conference on , 1998 , Page(s): 32 -39</p> |

- | | | | |
|-----|---|---|--|
| CNF |  |  | <p><u>Flow synchronization protocol</u>
 <i>Escobar, J.; Deutsch, D.; Patridge, C.</i>
 Global Telecommunications Conference, 1992. Conference Record., GLOBECOM '92. Communication for Global Users., IEEE , 1992 , Page(s): 1381 -1387 vol.3</p> |
| CNF |  |  | <p><u>Symbol-timing recovery for M-PSK modulation schemes using the signum function</u>
 <i>Verdin, D.; Tozer, T.C.</i>
 New Synchronisation Techniques for Radio Systems, IEE Colloquium on , 1995 , Page(s): 2/1 -2/7</p> |
| CNF |  |  | <p><u>A study of the dynamic governing timing restoration in the actively modelocked soliton laser</u>
 <i>Grein, M.E.; Jiang, L.A.; Haus, H.A.; Ippen, E.P.</i>
 Lasers and Electro-Optics, 1999. CLEO '99. Summaries of Papers Presented at the Conference on , 1999 , Page(s): 100</p> |
| CNF |  |  | <p><u>Linear timing acquisition scheme in DS-CDMA communication system</u>
 <i>Chao-Ming Chang; Kwang-Cheng Chen</i>
 Personal, Indoor and Mobile Radio Communications, 1998. The Ninth IEEE International Symposium on Volume: 3 , 1998 , Page(s): 1471 -1475 vol.3</p> |
| CNF |  |  | <p><u>Probability of error for fault-tolerant byte synchronization detectors</u>
 <i>Sobey, C.H.</i>
 Communications, 1996. ICC '96, Conference Record, Converging Technologies for Tomorrow's Applications. 1996 IEEE International Conference on Volume: 3 , 1996 , Page(s): 1528 -1532 vol.3</p> |
| CNF |  |  | <p><u>A synchronization scheme for distributed multimedia information systems</u>
 <i>Lu, G.J.; Pung, H.K.; Chua, T.S.</i>
 Singapore ICCS/ISITA '92. 'Communications on the Move' , 1990 , Page(s): 1 -5 vol.1</p> |
| PER |  |  | <p><u>MultiSync: a synchronization model for multimedia systems</u>
 <i>Herng-Yow Chen; Ja-Ling Wu</i>
 Selected Areas in Communications, IEEE Journal on Volume: 14 1 , Jan. 1996 , Page(s): 238 -248</p> |
| PER |  |  | <p><u>Optimizing synchronization in multiprocessor DSP systems</u>
 <i>Bhattacharyya, S.S.; Sriram, S.; Lee, E.A.</i>
 Signal Processing, IEEE Transactions on Volume: 45 6 , June 1997 , Page(s): 1605 -1618</p> |
| CNF |  |  | <p><u>A timing-driven soft-macro resynthesis method in interaction with chip floorplanning</u>
 <i>Hsiao-Pin Su; Wu, A.C.-H.; Youn-Long Lin</i>
 Design Automation Conference, 1999. Proceedings. 36th , 1999 , Page(s): 262 -267</p> |
| CNF |  |  | <p><u>Timing-driven HW/SW codesign based on task structuring and process timing simulation</u></p> |

Ramanathan, D.; Dasdan, A.; Gupta, R.
Hardware/Software Codesign, 1999. (CODES '99).
Proceedings of the Seventh International Workshop on , 1999
, Page(s): 203 -207

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 [Next]

| [IEL Online Home](#) | [Search](#) | [Advanced Search](#) | [What's New](#) | [Help](#) | [Logout](#) |
| [FAQ's](#) | [Support](#) | [Comments](#) |













Copyright 1999 Institute of Electrical and Electronics Engineers. All rights reserved.







































Collection: ☒ Journals ☒ Conferences ☒ Standards

Your search matched **714** of **608497** documents.

25 are presented on this page, sorted by Score in descending order.

DOC TYPE	VIEW ISSUE TOC	VIEW FULL PAGE	VIEW CITATION
CNF			<p><u>A probabilistic timing analysis for synthesis in microprocessor interface design</u> <i>Escalante, M.A.; Dimopoulos, N.J.</i> Communications, Computers, and Signal Processing, 1995. Proceedings., IEEE Pacific Rim Conference on , 1995 , Page(s): 277 -280</p>
PER			<p><u>Interface optimization for concurrent systems under timing constraints</u> <i>Filo, D.; Ku, D.C.; Coelho, C.N., Jr.; De Micheli, G.</i> Very Large Scale Integration (VLSI) Systems, IEEE Transactions on Volume: 13 , Sept. 1993 , Page(s): 268 -281</p>
CNF			<p><u>Timing analysis for synthesis of hardware interface controllers using timed signal transition graphs</u> <i>Escalante, M.A.; Dimopoulos, N.J.; Gyuroff, D.; Muller, H.</i> Petri Nets and Performance Models, 1995., Proceedings of the Sixth International Workshop on , 1995 , Page(s): 232 -240</p>
CNF			<p><u>Timing analysis for synthesis in microprocessor interface design</u> <i>Escalante, M.A.; Dimopoulos, N.J.</i> High-Level Synthesis, 1994., Proceedings of the Seventh International Symposium on , 1994 , Page(s): 23 -28</p>
CNF			<p><u>Interface timing verification with delay correlation using constraint logic programming</u> <i>Girodias, P.; Cerny, E.</i> European Design and Test Conference, 1997. ED&TC 97. Proceedings , 1997 , Page(s): 12 -19</p>
CNF			<p><u>Modeling data transfer signal timings in DAME</u> <i>Huber, B.; Li, K.; Dimopoulos, N.; Escalante, M.; Manning, E.</i> Communications, Computers and Signal Processing, 1993., IEEE Pacific Rim Conference on Volume: 2 , 1993 , Page(s): 505 -509 vol.2</p>

- | | | | |
|-----|---|---|---|
| CNF |  |  | <p><u>A design pattern based approach to generating synchronization adaptors from annotated IDL</u>
 <i>Jacobsen, H.-A.; Kramer, B.J.</i>
 Automated Software Engineering, 1998. Proceedings. 13th IEEE International Conference on , 1998 , Page(s): 63 -72</p> |
| CNF |  |  | <p><u>Representation of control and timing behavior with applications to interface synthesis</u>
 <i>Hayati, S.A.; Parker, A.C.; Granacki, J.J.</i>
 Computer Design: VLSI in Computers and Processors, 1988. ICCD '88., Proceedings of the 1988 IEEE International Conference on , 1988 , Page(s): 382 -387</p> |
| PER |  |  | <p><u>An expert system to automate timing design</u>
 <i>Kara, A.; Rastogi, R.; Kawamura, K.</i>
 IEEE Design & Test of Computers
 Volume: 5 5 , Oct. 1988 , Page(s): 28 -40</p> |
| CNF |  |  | <p><u>Performance analysis of the synchronization mechanism used at the VIRUS interface</u>
 <i>Maniatopoulos, A.; Antonakopoulos, T.; Makios, V.</i>
 Electronics, Circuits, and Systems, 1996. ICECS '96., Proceedings of the Third IEEE International Conference on Volume: 1 , 1996 , Page(s): 319 -322 vol.1</p> |
| CNF |  |  | <p><u>An algorithm for synthesis of system-level interface circuits</u>
 <i>Chung, K.-S.; Gupta, R.K.; Liu, C.L.</i>
 Computer-Aided Design, 1996. ICCAD-96. Digest of Technical Papers., 1996 IEEE/ACM International Conference on , 1996 , Page(s): 442 -447</p> |
| CNF |  |  | <p><u>A remote presentation agent for multimedia databases</u>
 <i>Rody, J.A.; Karmouch, A.</i>
 Multimedia Computing and Systems, 1995., Proceedings of the International Conference on , 1995 , Page(s): 223 -230</p> |
| PER |  |  | <p><u>The single-chip Fastbus slave interface</u>
 <i>Nelson, R.O.; Machen, D.R.; Downing, R.W.</i>
 Nuclear Science, IEEE Transactions on
 Volume: 37 2 2 , April 1990 , Page(s): 310 -314</p> |
| PER |  |  | <p><u>Source-synchronization and timing vernier techniques for 1.2-GB/s SDRAM interface</u>
 <i>Nakase, Y.; Morooka, Y.; Perlman, D.J.; Kolor, D.J.; Jae-Myoung Choi; Shin, H.J.; Yoshimura, T.; Watanabe, N.; Matsuda, Y.; Kumanoya, M.; Yamada, M.</i>
 Solid-State Circuits, IEEE Journal of
 Volume: 34 4 , April 1999 , Page(s): 494 -501</p> |
| PER |  |  | <p><u>Mixed digital/analog signal processing for a single-chip 2B1Q U-interface transceiver</u>
 <i>Batruni, R.; Lemaitre, P.; Fensch, T.</i>
 Solid-State Circuits, IEEE Journal of
 Volume: 25 6 , Dec. 1990 , Page(s): 1414 -1425</p> |
| CNF |  |  | <p><u>Tissue ischemia measurement using induced-current impedance spectroscopy: interface and timing considerations</u></p> |

- Petty, R.C.; Kun, S.; Peura, R.A.
 Bioengineering Conference, 1999. Proceedings of the IEEE
 25th Annual Northeast , 1999 , Page(s): 117 -119
- CNF  
An easy implementation of multimedia synchronization on a high-speed ring LAN
 Lee, W.-T.; Kung, L.-Y.; Liu, F.-H.
 Networks, 1993. International Conference on Information Engineering '93. 'Communications and Networks for the Year 2000', Proceedings of IEEE Singapore International Conference on
 Volume: 1 , 1993 , Page(s): 217 -220 vol.1
- PER  
An interface circuit for a Josephson-CMOS hybrid digital system
 Suzuki, M.; Maezawa, M.; Takato, H.; Nakagawa, H.; Hirayama, F.; Kiryu, S.; Aoyagi, M.; Sekigawa, T.; Shoji, A.
 Applied Superconductivity, IEEE Transactions on
 Volume: 9 2 3 , June 1999 , Page(s): 3314 -3317
- CNF  
Feedback-based synchronization for QoS traffic in cluster computing
 Hyo Jung Song; Chien, A.
 Parallel Interconnects, 1999. (PI '99). Proceedings. The 6th International Conference on , 1999 , Page(s): 205 -212
- CNF  
A front-end electronics module for the PHENIX pad chamber
 Smith, M.C.; Bryan, W.L.; Smith, D.; Jagadish, U.; McMillan, D.; Walker, J.; Young, G.R.; Oskarsson, A.; Osterman, L.; Greene, V.; Nikkinen, L.
 Nuclear Science Symposium, 1998. Conference Record. 1998 IEEE
 Volume: 2 , 1998 , Page(s): 910 -913 vol.2
- CNF  
Asynchronous interface specification, analysis and synthesis
 Kishinevsky, M.; Cortadella, J.; Kondratyev, A.
 Design Automation Conference, 1998. Proceedings , 1998 , Page(s): 2 -7
- CNF  
Considerations for implementing high performance VXL test systems
 Emmert, G.T.
 AUTOTESTCON '98. IEEE Systems Readiness Technology Conference., 1998 IEEE , 1998 , Page(s): 466 -473
- CNF  
Synthesis of interface controllers from timing diagram specifications
 El-Aboudi, A.; Aboulhamid, E.-M.; Cerny, E.
 Custom Integrated Circuits Conference, 1998. Proceedings of the IEEE 1998 , 1998 , Page(s): 89 -92
- CNF  
An on-chip timing adjuster with sub-100-ps resolution for a high-speed DRAM interface
 Noda, H.; Aoki, M.; Tanaka, H.; Nagahima, O.; Aoki, H.
 VLSI Circuits, 1998. Digest of Technical Papers. 1998 Symposium on , 1998 , Page(s): 62 -63

CNF

**A 2.6 GB/s multi-purpose chip-to-chip interface**

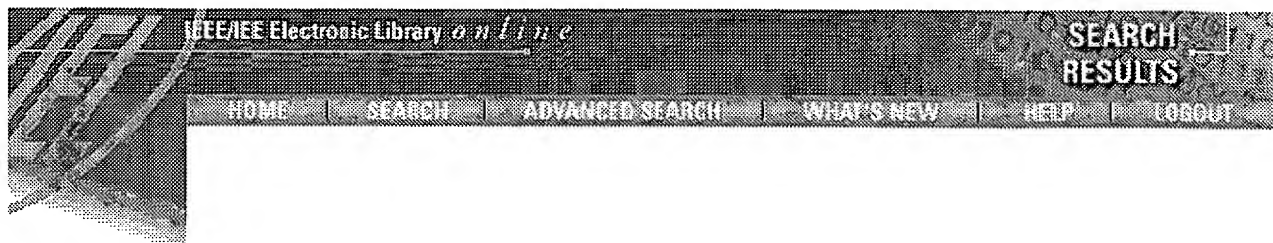
Lau, B.; Chan, Y.-F.; Moncayo, A.; Ho, J.; Allen, M.; Salmon, J.; Liu, J.; Muthal, M.; Lee, C.; Nguyen, T.; Horine, B.; Leddige, M.; Huang, K.; Wei, J.; Yu, L.; Tarver, R.; Hsia, Y.; Vu, R.; Tsern, E.; Liaw, H.-J.; Hudson, J.; Nguyen, D.; Donnelly, K.; Cr

Solid-State Circuits Conference, 1998. Digest of Technical Papers. 1998 IEEE International , 1998 , Page(s): 162 -163, 431

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [\[Next\]](#)

| [IEL Online Home](#) | [Search](#) | [Advanced Search](#) | [What's New](#) | [Help](#) | [Logout](#) |
| [FAQ's](#) | [Support](#) | [Comments](#) |

Copyright 1999 Institute of Electrical and Electronics Engineers. All rights reserved.



(synchronizing or timing or time adjustment) and

Collection: ☒ Journals ☒ Conferences ☒ Standards

Your search matched **1** of **608497** documents.

1 are presented on this page, sorted by Score in descending order.

DOC TYPE VIEW ISSUE TOC VIEW FULL PAGE VIEW CITATION

PER



A semi-custom protocol control logic device for FASTBUS

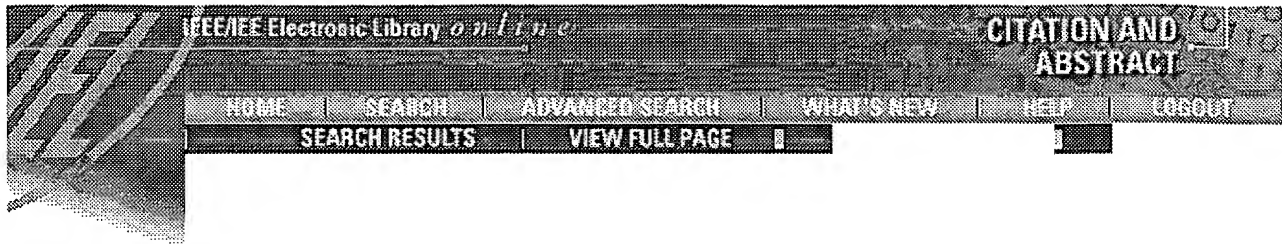
Skegg, R.; Daviel, A.

Nuclear Science, IEEE Transactions on

Volume: 35 1 1-2 , Feb. 1988 , Page(s): 306 -310

| [IEL Online Home](#) | [Search](#) | [Advanced Search](#) | [What's New](#) | [Help](#) | [Logout](#) |
| [FAQ's](#) | [Support](#) | [Comments](#) |

Copyright 1999 Institute of Electrical and Electronics Engineers. All rights reserved.



A semi-custom protocol control logic device for FASTBUS

- Skegg, R.; Daviel, A.

TRIUMF, Vancouver, BC, Canada

This Paper Appears in :
Nuclear Science, IEEE Transactions on

on Pages: 306 - 310

Feb. 1988 Vol. 35 Issue: 1 Part: 1-2 ISSN: 0018-9499

References Cited: 3

CODEN: IETNAE

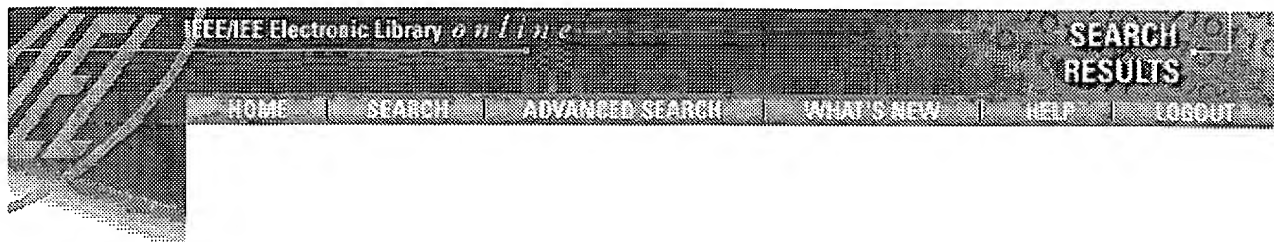
Accession Number: 3212308

Abstract:

A gate-array device called the protocol control logic (PCL) is presented that greatly simplifies the construction of a FASTBUS slave interface. This device connects to the segment timing and control signals and together with address/data interface chips produces a simple set of signals for the user. The entire FASTBUS protocol is accommodated. Great flexibility of response is provided by many control inputs to the PCL. These allow the suppression of features that may not be desired in a particular implementation or the generation of special responses. The PCL produces several status signals to aid the user in detecting conditions of interest.

Subject Terms:

gate-array device; protocol control logic; FASTBUS slave interface; segment timing; control signals; FASTBUS protocol; cellular arrays; computer interfaces; protocols



Collection: ☒ Journals ☒ Conferences ☒ Standards

Your search matched **133** of **608497** documents.

25 are presented on this page, sorted by Score in descending order.

DOC TYPE	VIEW ISSUE TOC	VIEW FULL PAGE	VIEW CITATION
CNF			<p><u>An algorithm for synthesis of system-level interface circuits</u> Chung, K.-S.; Gupta, R.K.; Liu, C.L. Computer-Aided Design, 1996. ICCAD-96. Digest of Technical Papers., 1996 IEEE/ACM International Conference on , 1996 , Page(s): 442 -447</p>
PER			<p><u>An interface circuit for a Josephson-CMOS hybrid digital system</u> Suzuki, M.; Maezawa, M.; Takato, H.; Nakagawa, H.; Hirayama, F.; Kiryu, S.; Aoyagi, M.; Sekigawa, T.; Shoji, A. Applied Superconductivity, IEEE Transactions on Volume: 9 2 3 , June 1999 , Page(s): 3314 -3317</p>
PER			<p><u>The single-chip Fastbus slave interface</u> Nelson, R.O.; Machen, D.R.; Downing, R.W. Nuclear Science, IEEE Transactions on Volume: 37 2 2 , April 1990 , Page(s): 310 -314</p>
CNF			<p><u>Interface timing verification with delay correlation using constraint logic programming</u> Girodias, P.; Cerny, E. European Design and Test Conference, 1997. ED&TC 97. Proceedings , 1997 , Page(s): 12 -19</p>
CNF			<p><u>Logic reduction in timed asynchronous circuits</u> Baake, U.; Huss, S.A. Circuits and Systems, 1995. ISCAS '95., 1995 IEEE International Symposium on Volume: 2 , 1995 , Page(s): 1223 -1226 vol.2</p>
CNF			<p><u>Test for designability-a new architecture for VLSI testers</u> Scott, M.A. Design for Testability, IEE Colloquium on , 1991 , Page(s): 9/1 -9/4</p>
CNF			<p><u>Bridging modularity and optimality: delay-insensitive interfacing in asynchronous circuits synthesis</u></p>

Saito, H.; Kondratyev, A.; Cortadella, J.; Lavagno, L.; Yakovlev, A.
Systems, Man, and Cybernetics, 1999. IEEE SMC '99
Conference Proceedings. 1999 IEEE International Conference
on
Volume: 3 , 1999 , Page(s): 899 -904 vol.3

CNF



RTL modeling of the RCMP egress routing logic

Palanisamy, M.; Tahar, S.

Electrical and Computer Engineering, 1999 IEEE Canadian
Conference on

Volume: 1 , 1999 , Page(s): 111 -115 vol.1

CNF



**110 GB/s simultaneous bi-directional transceiver logic
synchronized with a system clock**

*Takahashi, T.; Muto, T.; Shirai, Y.; Shirotori, F.; Takada, Y.;
Yamagiwa, A.; Nishida, A.; Kiyuna, T.*

Solid-State Circuits Conference, 1999. Digest of Technical
Papers. ISSCC. 1999 IEEE International , 1999 , Page(s): 176
-177

CNF



**1998 IEEE/ACM International Conference on Computer-Aided
Design. Digest of Technical Papers (IEEE Cat. No.98CB36287)**

Computer-Aided Design, 1998. ICCAD 98. Digest of
Technical Papers. 1998 IEEE/ACM International Conference
on , 1998

CNF



An ASIC library granular DRAM macro with built-in self test

Dreibelbis, J.; Barth, J., Jr.; Kho, R.; Kalter, T.

Solid-State Circuits Conference, 1998. Digest of Technical
Papers. 1998 IEEE International , 1998 , Page(s): 74 -75, 416

CNF



A flexible compatible PCI interface for nuclear experiments

Saleh, H.; Engels, R.; Reinartz, R.; Reinhart, P.; Rongen, F.

Nuclear Science Symposium, 1997. IEEE , 1997 , Page(s):
704 -706 vol.1

CNF



**Formal implementation verification of the bus interface unit for
the Alpha 21264 microprocessor**

Bischoff, G.P.; Brace, K.S.; Jain, S.; Razdan, R.

Computer Design: VLSI in Computers and Processors, 1997.
ICCD '97. Proceedings., 1997 IEEE International Conference
on , 1997 , Page(s): 16 -24

CNF



**Interface techniques for embedding custom mega cells in a
gate array**

Ashby, L.R.

Custom Integrated Circuits Conference, 1993., Proceedings of
the IEEE 1993 , 1993 , Page(s): 23.5.1 -23.5.4

CNF



**Proceedings of IEEE Custom Integrated Circuits Conference -
CICC '93**















Custom Integrated Circuits Conference, 1993., Proceedings of
the IEEE 1993 , 1993

CNF



**Single-chip 1062 Mbaud CMOS transceiver for serial data
communication**

Ewen, J.F.; Widmer, A.X.; Soyuer, M.; Wrenner, K.R.; Parker, B.;

- | | | | |
|-----|---|---|---|
| | | | <i>Ainspan, H.A.</i>
Solid-State Circuits Conference, 1995. Digest of Technical Papers. 41st ISSCC, 1995 IEEE International , 1995 ,
Page(s): 32 -33, 336 |
| CNF |  |  | <u>Rethinking system-level development: striving for coherency, certitude, and realizability</u>
<i>Mulcare, D.B.</i>
Systems Engineering of Computer Based Systems, 1995.,
Proceedings of the 1995 International Symposium and
Workshop on , 1995 , Page(s): 143 -150 |
| CNF |  |  | <u>Rapid prototyping of DSP systems via system interface module generation</u>
<i>Famorzadeh, S.; Madisetti, V.K.</i>
Acoustics, Speech, and Signal Processing, 1996. ICASSP-96.
Conference Proceedings., 1996 IEEE International
Conference on
Volume: 2 , 1996 , Page(s): 1256 -1259 vol. 2 |
| CNF |  |  | <u>Vexcel corporations' SAR processing ground station: An overview</u>
<i>Compton, M.; Carande, R.E.</i>
Geoscience and Remote Sensing Symposium, 1996. IGARSS
'96. 'Remote Sensing for a Sustainable Future.', International
Volume: 2 , 1996 , Page(s): 1211 -1213 vol.2 |
| CNF |  |  | <u>Upgrade of NSLS timing system</u>
<i>Singh, O.; Ramamoorthy, S.; Sheehan, J.; Smith, J.</i>
Particle Accelerator Conference, 1995., Proceedings of the
1995
Volume: 4 , 1995 , Page(s): 2223 -2225 vol.4 |
| CNF |  |  | <u>Multimedia bridging: a structured view</u>
<i>Krishnaswamy, V.; Thuel, S.R.</i>
Emerging Technologies and Applications in Communications,
1996. Proceedings., First Annual Conference on , 1996 ,
Page(s): 144 -147 |
| CNF |  |  | <u>An approach for a dynamic generation/validation system for the functional simulation considering timing constraints</u>
<i>Heinkel, U.; Glauert, W.H.</i>
European Design and Test Conference, 1996. ED&TC 96.
Proceedings , 1996 , Page(s): 302 -306 |
| CNF |  |  | <u>Transformation of timing diagram specifications into VHDL code</u>
<i>Grass, W.; Grobe, C.; Lenk, S.; Tiedemann, W.-D.; Kloos, C.D.;
Marin, A.; Robles, T.</i>
Design Automation Conference, 1995. Proceedings of the
ASP-DAC '95/CHDL '95/VLSI '95., IFIP International
Conference on Hardware Description Languages. IFIP
International Conference on Very Large Scale Integration.,
Asian and South Pacific , 1995 , Page(s): 659 -668 |

CNF

**Assessing the feasibility of interface designs before their implementation***Escalante, M.A.; Dimopoulos, N.J.*

Design Automation Conference, 1995. Proceedings of the ASP-DAC '95/CHDL '95/VLSI '95., IFIP International Conference on Hardware Description Languages. IFIP International Conference on Very Large Scale Integration., Asian and South Pacific , 1995 , Page(s): 149 -154

CNF

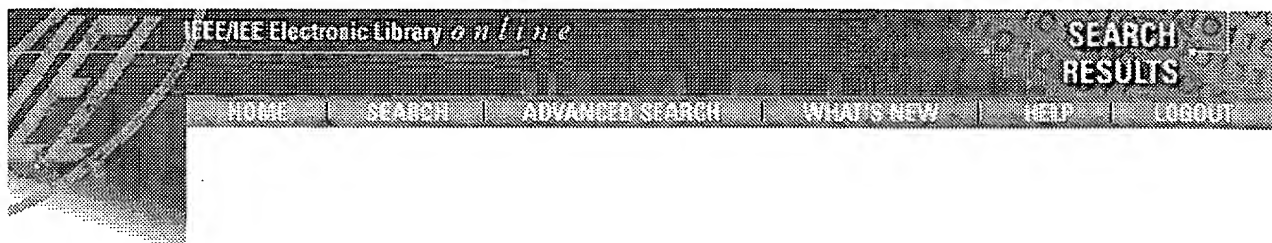
**A hybrid real-time UNIX controller for the Donner 600-crystal positron tomograph***Huesman, R.H.; Klein, G.J.; Fleming, T.K.*

Nuclear Science Symposium and Medical Imaging Conference, 1994., 1994 IEEE Conference Record Volume: 4 , 1995 , Page(s): 1648 -1651 vol.4

1 2 3 4 5 6 [Next]

| [IEL Online Home](#) | [Search](#) | [Advanced Search](#) | [What's New](#) | [Help](#) | [Logout](#) |
| [FAQ's](#) | [Support](#) | [Comments](#) |

Copyright 1999 Institute of Electrical and Electronics Engineers. All rights reserved.







(synchronizing or timing or time adjustment) and

Collection: ☒ Journals ☒ Conferences ☒ Standards

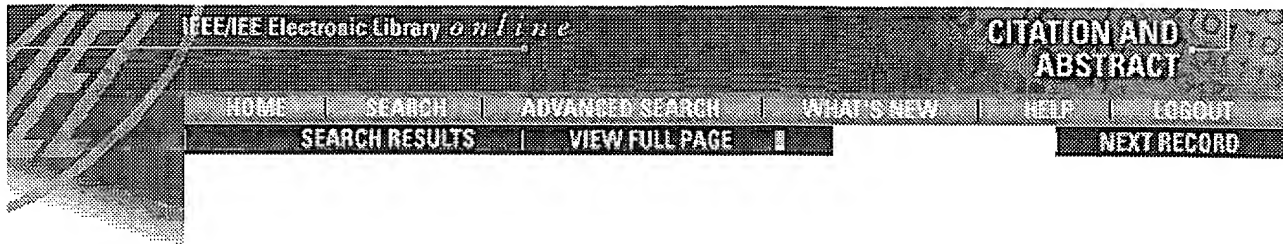
Your search matched **2** of **608497** documents.

2 are presented on this page, sorted by Score in descending order.

DOC TYPE	VIEW ISSUE TOC	VIEW FULL PAGE	VIEW CITATION
CNF			<p><u>Scalable Coherent Interface</u> Alnaes, K.; Kristiansen, E.H.; Gustavson, D.B.; James, D.V. CompuEuro '90. Proceedings of the 1990 IEEE International Conference on Computer Systems and Software Engineering , 1990 , Page(s): 446 -453</p>
PER			<p><u>Trigger and data acquisition system for the N-N experiment</u> Baldo-Ceolin, M.; Benetti, P.; Bobisut, F.; Bonaiti, V.; Calligarich, E.; Cavestro, A.; Dolfini, R.; Genoni, M.; Gibin, D.; Berzolari, A.G.; Guglielmi, A.; Mattioli, F.; Mauri, F.; Mezzetto, M.; Piazzoli, A.; Puglierin, G.; Rappoldi, A.; Raselli, G.L.; Sc Nuclear Science, IEEE Transactions on Volume: 38 2 1-2 , April 1991 , Page(s): 471 -475</p>

| [IEL Online Home](#) | [Search](#) | [Advanced Search](#) | [What's New](#) | [Help](#) | [Logout](#) |
| [FAQ's](#) | [Support](#) | [Comments](#) |

Copyright 1999 Institute of Electrical and Electronics Engineers. All rights reserved.



Scalable Coherent Interface

- Alnaes, K.; Kristiansen, E.H.; Gustavson, D.B.; James, D.V.

Dolphin Server Technol. AS, Oslo, Norway

This Paper Appears in :

CompuEuro '90. Proceedings of the 1990 IEEE International Conference on Computer Systems and Software Engineering

on Pages: 446 - 453

This Conference was Held : 8-10 May 1990

1990

ISBN: 0-8186-2041-2

Total Pages: xiii+574

References Cited: 11

Accession Number: 3842745

Abstract:

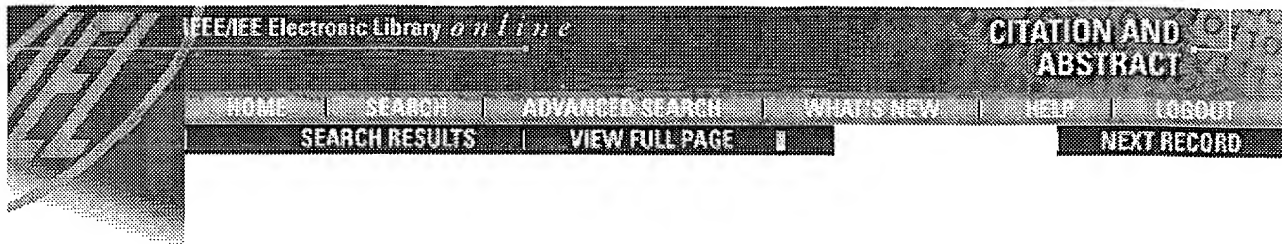
The Scalable Coherent Interface Project (IEEE P1596) is establishing an interface standard for very-high-performance multiprocessors, supporting a cache-coherent-memory model scalable to systems with up to 64K nodes. The P1596 Scalable Coherent Interface (SCI) will supply a peak bandwidth per node of 1 Gb/s. The SCI standard should facilitate assembly of processor, memory, I/O and bus bridge cards from multiple vendors into massively parallel systems with throughput far above what is possible today. The SCI standard encompasses two levels of interface, a physical level and a logical level. The physical level specifies electrical, mechanical and thermal characteristics of connectors and cards that meet the standard. The logical-level describes the address space, data transfer protocols, cache coherence mechanisms, synchronization primitives and error recovery. Logical-level issues such as packet formats, packet transmission, transaction handshake, flow control, and cache coherence are addressed.

Subject Terms:

electrical characteristics; mechanical characteristics; Scalable Coherent Interface Project; IEEE P1596; interface standard; very-high-performance multiprocessors; cache-coherent-memory model; peak bandwidth; massively parallel systems; SCI standard; physical level; logical level; thermal characteristics; connectors; cards; address space; data transfer protocols; cache coherence mechanisms; synchronization primitives; error recovery; packet formats; packet transmission; transaction handshake; flow control; 1 Gbit/s; buffer storage; computer interfaces; multiprocessing systems; standards

| [IEL Online Home](#) | [Search](#) | [Advanced Search](#) | [What's New](#) | [Help](#) | [Logout](#) |
| [FAQ's](#) | [Support](#) | [Comments](#) |

Copyright 1999 Institute of Electrical and Electronics Engineers. All rights reserved.



Scalable Coherent Interface

- Alnaes, K.; Kristiansen, E.H.; Gustavson, D.B.; James, D.V.
Dolphin Server Technol. AS, Oslo, Norway

This Paper Appears in :

CompuEuro '90. Proceedings of the 1990 IEEE International Conference on Computer Systems and Software Engineering

on Pages: 446 - 453

This Conference was Held : 8-10 May 1990

1990

ISBN: 0-8186-2041-2

Total Pages: xiii+574

References Cited: 11

Accession Number: 3842745

Abstract:

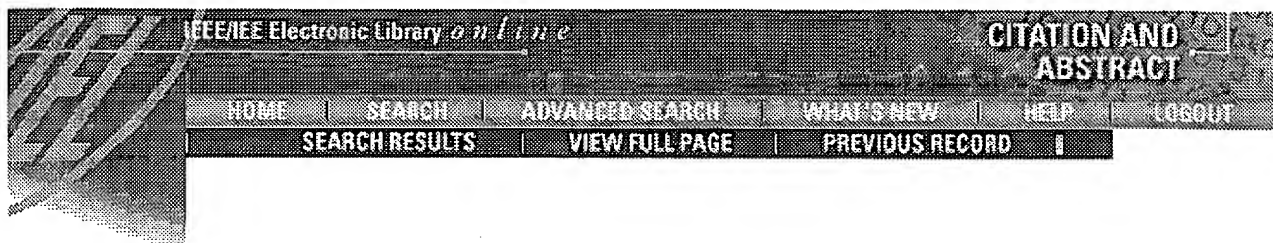
The Scalable Coherent Interface Project (IEEE P1596) is establishing an interface standard for very-high-performance multiprocessors, supporting a cache-coherent-memory model scalable to systems with up to 64K nodes. The P1596 Scalable Coherent Interface (SCI) will supply a peak bandwidth per node of 1 Gb/s. The SCI standard should facilitate assembly of processor, memory, I/O and bus bridge cards from multiple vendors into massively parallel systems with throughput far above what is possible today. The SCI standard encompasses two levels of interface, a physical level and a logical level. The physical level specifies electrical, mechanical and thermal characteristics of connectors and cards that meet the standard. The logical-level describes the address space, data transfer protocols, cache coherence mechanisms, synchronization primitives and error recovery. Logical-level issues such as packet formats, packet transmission, transaction handshake, flow control, and cache coherence are addressed.

Subject Terms:

electrical characteristics; mechanical characteristics; Scalable Coherent Interface Project; IEEE P1596; interface standard; very-high-performance multiprocessors; cache-coherent-memory model; peak bandwidth; massively parallel systems; SCI standard; physical level; logical level; thermal characteristics; connectors; cards; address space; data transfer protocols; cache coherence mechanisms; synchronization primitives; error recovery; packet formats; packet transmission; transaction handshake; flow control; 1 Gbit/s; buffer storage; computer interfaces; multiprocessing systems; standards

| [IEL Online Home](#) | [Search](#) | [Advanced Search](#) | [What's New](#) | [Help](#) | [Logout](#) |
| [FAQ's](#) | [Support](#) | [Comments](#) |

Copyright 1999 Institute of Electrical and Electronics Engineers. All rights reserved.



Trigger and data acquisition system for the N-N experiment

- Baldo-Ceolin, M.; Benetti, P.; Bobisut, F.; Bonaiti, V.; Calligarich, E.; Cavestro, A.; Dolfini, R.; Genoni, M.; Gibin, D.; Berzolari, A.G.; Guglielmi, A.; Mattioli, F.; Mauri, F.; Mezzetto, M.; Piazzoli, A.; Puglierin, G.; Rappoldi, A.; Raselli, G.L.; Scannicchio, D.; Schiavon, P.G.; Sconza, A.; Vascon, M.; Visentin, L.
Dipartimento di Fisica G. Galilei, Padova Univ., Italy

This Paper Appears in :
Nuclear Science, IEEE Transactions on

on Pages: 471 - 475

April 1991 Vol. 38 Issue: 2 Part: 1-2 ISSN: 0018-9499

References Cited: 11

CODEN: IETNAE

Accession Number: 4004066

Abstract:

The trigger and data acquisition system for the N-N experiment at the Institute Laue-Langevin at Grenoble is described along with CAMAC modules especially designed for this experiment. The trigger system is organized around three logical levels; it works in the presence of a high level of beam-induced noise, without beam pulse synchronization, looking for a very rare signal. The final trigger rate is 4 Hz, not very different from the beam off rate, while the trigger efficiency for the antineutron annihilation detection is < 0.7 . The data acquisition is based on a MicroVAX II computer, in a cluster with four VaxStations, interfaces to CAMAC, and uses a modified version of the DAQP software developed at CERN. The system, has been working for a year with high efficiency and reliability.

Subject Terms:

Subject Terms:

trigger system; data acquisition system; CAMAC modules; trigger rate;
trigger efficiency; antineutron annihilation; MicroVAX II computer;
DAQP software; 4 Hz; CAMAC; data acquisition

[| IEL Online Home |](#) [Search |](#) [Advanced Search |](#) [What's New |](#) [Help |](#) [Logout |](#)
[| FAQ's |](#) [Support |](#) [Comments |](#)

Copyright 1999 Institute of Electrical and Electronics Engineers. All rights reserved.

A SEMI-CUSTOM PROTOCOL CONTROL LOGIC DEVICE FOR FASTBUS

Robert Skegg
Andrew Daviel

TRIUMF, Vancouver, B.C., CANADA, V6T 2A3.

Abstract

A gate array device called the Protocol Control Logic (PCL) is presented which greatly simplifies the construction of a FASTBUS Slave interface. This device connects to the Segment timing and control signals and together with Address/Data Interface chips produces a simple set of signals for the user. The entire FASTBUS protocol is accommodated.

Great flexibility of response is provided by many control inputs to the PCL. These allow the suppression of features which may not be desired in a particular implementation, or the generation of special responses. Additionally the PCL produces several status signals to aid the user in detecting conditions of interest.

Introduction

FASTBUS (IEEE Std 960-1986 [1]) provides a physical and electrical standard for the interconnection of Master and Slave devices to a standard bus, called a Segment. Due to the large number of modes of operation available within the specification, interfacing a Slave to a FASTBUS Segment is a complex task if more than the most basic features of the specification are desired. In order to simplify the task of designing Slave and to some extent Master interfaces a set of semi-custom circuits was proposed. The first of these devices, the Address/Data Interface, was described at the 1984 IEEE NSS [2]. A pair of ADIs provides a 32-bit wide Address/Data de-multiplexer, various standard registers, primary address detection and parity logic. The ADI is now commercially available [3].

In order to control the ADIs a second gate-array device is being designed. The Protocol Control Logic (PCL) responds to Segment control and timing signals, and provides signals to the ADIs as well as simple strobes for the front-end of the Slave module. Using these devices it is easy to design a Slave which responds to all FASTBUS modes of operation.

Functions of the PCL

Figure 1 shows the major parts of the PCL. It contains an Address Cycle section and a Data Cycle section. Figure 2 shows the general interconnection of the PCL, ADIs, Segment ECL/TTL translator/transceivers and the user's registers.

All defined addressing modes are supported by the PCL and ADIs. Geographic, Logical and Class N Broadcast Primary address recognition is performed by the address detectors in the ADIs and the results passed to the PCL. Other Broadcast modes are identified within the PCL. The bus timing signal AS is used by the PCL to initiate attaching procedures. For non-broadcast addresses AK is produced. For Logical addresses to data space the NTA register in the ADIs is loaded with the Internal Address.

The PCL is capable of performing all Data Cycles, including the proposed enhancements (MS4,6,7). The user provides signals to the PCL from an NTA decoder indicating the location and type of register addressed. SS responses are generated for indicated bad address or parity errors. The user may disable automatic SS code generation and/or provide an external SS code. An external SS code of 6 will prevent the PCL from executing any data cycle. The user may also indicate a busy condition (giving SS=1) or bad data (giving SS=7). The user is provided with convenient control signals for CSR#0 and user registers.

Parity for the AD field is generated by the ADIs. During Write operations the PCL combines this information with the Segment Parity Enable and Parity signals and takes appropriate action. During Read operations PA and PE are asserted on the Segment. The parity generation logic in the ADIs is not very fast. Including the logic required in the PCL, it takes typically 80 ns to safely check or generate PA. This slows the operation of the interface. If a user wishes to ignore parity checking, the PA and PE Segment signals may be ignored and the BPA pin of the PCL tied to 0V through a 1k ohm resistor.

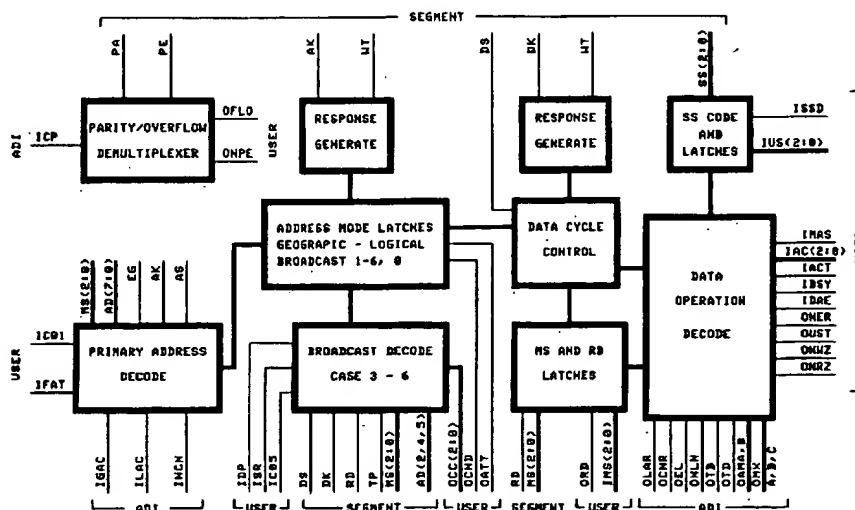


Figure 1. Block Diagram of the protocol Control Logic.

The PCL responds automatically to the Segment Wait signal. The user may assert the WT signal which will control both his PCL and other FASTBUS devices.

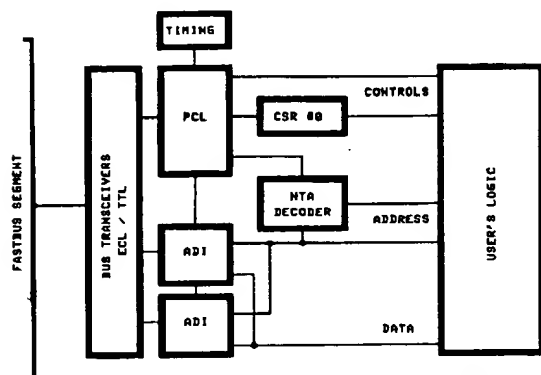


Figure 2. Block Diagram of a Slave Interface.

Timing of Operations

FASTBUS is asynchronous in operation. There is no common clock to time all operations. Instead there are strobe lines from the Master to start operations and acknowledge lines from the Slave to indicate completion. These signals are AS and AK for address operations and DS and DK for data operations. This system allows the Master and Slave to proceed at their own speed.

Within each operation a Slave interface must perform many steps, for example: decoding the type of operation; checking or generating parity; loading a register. Each step must be timed correctly and the acknowledge response sent to the Master only when the response is stabilized on the Segment. For an interface capable of responding to all FASTBUS operations this is a complex task.

The PCL employs five external delays to correctly time all operations. Because the maximum allowed Slave Response Time is 1 micro-second simple R/C integrators may be used. The PCL inputs from the delays are of the Schmidt trigger type. Two of the delays act upon the Segment timing signals AS and DS. These allow for internal delays in decoding the MS codes associated with the operation. Two more delay elements are used by the PCL to time the steps of each operation. One has a relatively long delay and is used to allow for address and parity decode times. The other has a short delay and is used to allow outputs of latches to settle. Both the rising and falling edge delays are used. The exact use of each delay period depends on the type of operation being performed. The fifth delay is used after internal address changes to accommodate user memory having a long address-data access time.

Pipelined Structure

The ADIs contain a data latch which is used as a Protective Buffer and also to temporarily hold data between the Segment and the user. Data operations may therefore be pipelined through the PCL and ADIs. (This is not to be confused with the MS-3 "Pipelined Block Transfer" operation.) In each operation the PCL returns the acknowledge signal to the Master as soon as it has captured Write data or is asserting Read data and has determined the appropriate SS response code. The PCL also saves the current MS and RD codes, and completes the operation whilst the Master sets up the next. Should the next strobe arrive before the PCL has completed an operation, it recognises and saves the strobe, and proceeds with the new operation as soon as the last one is completed.

PCL Connections

Connections to the Segment, timers and the ADIs are straightforward, and shown in Figure 3, an application example. Signals to and from the user are summarised below.

PCL User Signals

IACT Accept data having parity error.
IAC[2:0] Output of NTA decoder.
IBSY Make Busy.
IC01 Enable Logical Addressing. (CSR#0<1>)
IC05 SR Flag. (CSR#0<5>)
IC7R Force Clear of CSR#7 in ADIs.
IDP Data present signal for Broadcast case 3/3a.
IFAT Force attach, for Broadcast case 8.
IMAS Indicates device is current Master.
INIT Initialize PCL.
ISR User's service request, for Broadcast case 5.
ISSD Disable internal SS generation.
IUS[2:0] User's SS code input.
OATT Indicates Slave is attached.
OCC[3:0] Attach method code.
OCND Latched CSR/not DSR.
OFLO NTA overflow flag.
OMS[2:0] Latched MS for data cycle.
ONPE Parity error flag, active low.
ONRZ Read CSR#0 gate, active low.
ONWZ Write CSR#0 gate, active low.
ORD Latched RD.
OUST User register data strobe.

Signals To The User

OUST is a timing signal that goes high to indicate that user RAM, FIFO registers or other externally implemented registers (but NOT CSR#0) should either be written or read, depending on the state of ORD.

OCND indicates if the current operation is to Control or Data space (High - Control space). This is the latched state of MS0 received at a recognised Primary Address time.

OMS(2:0) are the latched MS signals and ORD is the latched Read signal. At the start of each Data transfer operation by the PCL the state of MS(2:0) and RD are latched, and are presented to the user for NTA decoding and any other purpose.

OFLO is effectively the 33rd bit of the NTA register. If the NTA register is incremented when it was set to FFFFFFFF hex it will go to 0 and OFLO will be set. The next operation of any kind on the NTA register will cause OFLO to go low. This output may be used by the NTA decoder to detect an error condition during Block Transfers to a device which has a legitimate NTA of FFFFFFFF hex.

OATT goes high when the Slave becomes attached. OCC(3:0) then indicates the method by which the Master attached to the Slave. The code used is shown below.

OCC3	OCC2	OCC1	OCC0	Mode
0	0	0	0	Broadcast Case 1
0	0	0	1	Broadcast Case 2
0	0	1	0	Broadcast Case 3
0	0	1	1	Broadcast Case 3a
0	1	0	0	Broadcast Case 4
0	1	0	1	Broadcast Case 5
0	1	1	0	Broadcast Case 6
0	1	1	1	Broadcast Case 7
1	0	0	0	Geographical
1	0	0	1	Logical
Remainder not used.				

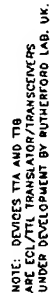


Figure 3. Slave design used for simulation and testing.

CSR#0

The user must provide the CSR#0 register. The exact makeup of this register depends on the features of the Slave. The PCL provides specific strobes to read (ONRZ) and write (ONWZ) CSR#0 which are "active low" to allow the use of the 74LS112 JK flip flop for the register and the 74LS244 tristate buffer to read it.

IC01 is connected to CSR#0<01>, the Logical Addressing enable bit. The PCL checks that this input is high before attaching upon recognition of a Logical Address.

Initialization and Reset

INIT initializes various latches within the PCL. It should be driven high by the power-on-clear circuit of the Slave for at least 200ns then held low.

IC7R clears the Class N register in the ADIS. It should be driven by the power-on-clear circuit of the Slave and also by CSR#0 bit 30. A high condition for 200ns or more will perform the reset.

NTA Decoding

The user is required to provide an internal NTA decoder which examines the output of the NTA register (ADI OA(31:00) and/or OACA, OACB), ORD and OCND. The decoder indicates to the PCL if the address is valid, where it is located and whether it is RAM or FIFO memory. This information is encoded on IAC(2:0) as shown below.

LOCATION	IAC(2:0)
CSR#0	0
User RAM	1
Not Used	2
CSR#3	3
CSR#4	4
User FIFO	5
Not Used	6
CSR#7	7

A simple Slave which has only RAM type user locations and CSRs#0 and 3 may tie IAC2 low and provide just two signals from the NTA decoder to IAC(1:0). A more complex Slave with FIFO locations, or CSR#4 or 7 must drive all three inputs. Addresses which are not implemented must produce the 'Not Used' code.

The effect of indicating that a register is a FIFO prevents incrementing of the NTA during Block and Pipeline transfers. To indicate the end of data the user must change IAC(2:0) to 'Not Used'.

If a 'Not Used' address is arrived at by means of a Block or Pipelined Transfer, then following Block or Pipelined transfers will produce an SS-2. If a Single Transfer is attempted to or from a 'Not Used' address, an SS-6 is produced. If a Block or Pipelined operation with a 'Not Used' address is attempted after a Single Transfer or Secondary Address operation, then an SS-6 is generated.

The CSR#4 code uses the 'class' register (CSR#7) in a 'Device User Address Register' mode. If this application is preferred in place of Class N Broadcast addressing, the CSR#7 code should not be used, and the ADI ONCN line tied high.

The NTA decoder also provides controls for the user's registers. These are gated by the timing signal OUST.

Responding with Busy

If the Slave contains locations which cannot respond to an operation within the response time of the Segment, the user may assert IBSY to the PCL. SS-1 will then be sent to the Master, and no further action will be taken by the interface.

Service Requests

IC05 is connected to CSR#0<05> and ISR is connected to the signal driving the SR Segment driver. The PCL uses these signals to generate the T pin responses to Broadcast Cases 5 and 6.

Special Primary Address Signals

IDP is the "data present" signal which is used in Broadcast Cases 3 and 3a to control the T pin response. IDP is exclusive-ORed within the PCL with the saved condition of AD(4) of a Broadcast Address Cycle in order to provide both Case 3 and 3a operation. If the user has a "device available" condition which differs from "no data present" then extra external gating will be required before IDP to achieve the desired responses.

IFAT when pulsed high will cause the Slave to immediately become attached in the Broadcast mode (no segment responses generated) to the current Master. It will stay attached until AS(d). This may be used to implement Broadcast Case 8 (manufacturer's mode). The user must examine AS, AK, MS(2,1) and the AD lines, and when the appropriate pattern is detected apply a 0 to 1 transition to IFAT.

Parity Checking

The PCL has a standard response of SS-6 (and will not accept the data) to a parity error during a Write Data Cycle. The Master is expected to retry the operation to correct the error. If the Master does not retry in a Block Transfer then all following data in the Block Transfer will be mis-located. An SS-6 response during a Pipelined Transfer will indicate that all following data is mis-located.

The user may opt to respond with SS-7, accept the bad data and thus maintain Block or Pipelined data synchronism by setting IACT high for any particular address and MS combination. This signal is best generated by the user's NTA decoder.

ONPE goes low if a parity error is detected during a Write operation. It may be used to set CSR#0<14>, the Parity Error Bit.

Use in a Master

Any FASTBUS Master device must have a Slave part. Certain CSR locations are mandatory for Masters. When used in the Slave part of a Master, the PCL allows normal Segment access to the Slave registers within the Master. When the Master gains mastership of the bus, it pulls IMAS high. The PCL will then use the ADIS to generate and check parity of the AD signals for the Master. The result of the check is indicated on ONPE. This should be externally gated by /IMAS in order not to set CSR#0<13> in the Slave part when the error is due to reading some other slave.

The Master may also address its own Slave part. Under these circumstances parity is neither generated nor checked, but all other signals are presented to the Segment to allow testing by snoop modules.

Slave Status Code Generation

SS codes are generated by the PCL according to the priority shown below.

LEVEL (highest)	SS	USE
0	IUS	User's code applied to IUS[2:0]
1	0	If ISSD-1
2	1	IBSY asserted
3	7	Sec. Addr. R/W with bad NTA or parity error
3	6	Single data op with bad NTA
3	2	Block or Pipelined op with bad NTA
4	6	Parity error, IACT-0
4	7	Parity error, IACT-1

When a code other than normal is required, the user may place the required pattern on IUS(2:0). To prevent this pattern being ORed with the internal response asserting any IUS input or ISSD will suppress the internally generated SS code. With the following exceptions, asserting IUS codes will not change the other actions of the PCL.

If it is desired not to respond to a particular operation, for example MS=7, the operation may be detected and IUS=6 applied to the PCL. The operation will then be acknowledged with an SS=6 response and the normal actions for the operation will not be carried out.

If ISSD is asserted and IUS=6, the PCL will respond to the operation with SS=0 and normal operation will not be carried out. This may be used, for example, to modify the MS=4 response from the proposed Single data transfer to a "delimiting action."

In either of these modified operations the ADI data latches will still be updated with any data that may otherwise have been transferred. This would only be apparent on an immediately following Read Protective Buffer operation.

Simulation and Testing

The PCL is being designed on a Daisy Personal Logician work station. In order to test the PCL it has been incorporated in a hypothetical FASTBUS slave with two ADIs, a RAM, CSR#0 and a simulated FIFO-like device [Fig. 3]. The resulting Slave is then exercised using a FASTBUS list processor written in a behavioural modelling language. This processor also incorporates EG generation and broadcast handshake logic.

The Master is a fixed program which reads a list of operations from a ROM rather than from the program. Lists of FASTBUS operations are written in a crude assembly language, then compiled for loading into the ROM. The Master then performs the operations in an asynchronous manner. Various parameters are settable interactively away from initial default values, such as AS and DS skew time, DK-DS turnaround time, timeouts, etc. Since this Master is pure software, it can be made as fast or as slow as desired to exercise the Slave at extremes of speed. While running, the Master displays each operation on the simulator screen, together with any errors encountered (SS error, timeout, parity, unexpected data on read). This output reduces the need to manually check simulated waveforms.

Production

The PCL is in the final stages of development. There remains a considerable amount of testing and circuit refinement to be done. The ADI was manufactured using a low power Schottky gate array. Advances in CMOS technology are now providing a challenge to bipolar arrays in both speed and price, therefore these arrays are now being investigated for performance and compatibility with the ADI. The packaging of the PCL will be an 88 lead pin grid array, which is similar to the ADI.

Conclusion

Much work has been done on the design of the PCL. International cooperation regarding the functionality of the PCL and the FASTBUS specification has produced a design for a Slave Interface providing all the features of FASTBUS. It is hoped that the PCL will be in production early in 1988.

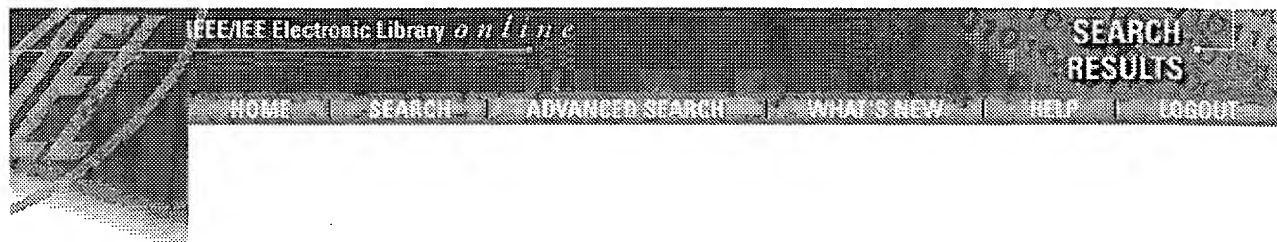
Acknowledgements

The authors wish to thank the members of the FASTBUS Working Groups for their assistance in the development of the PCL.

This work was supported by the National Research Council of Canada.

References



- [1] FASTBUS. Modular High Speed Data Acquisition and Control System, IEEE Std 960-1986 with Addenda and Errata of May 1987, U.S. NIM Committee.
- [2] R. Skegg and A. Daviel, A General Purpose FASTBUS Interface Chipset, IEEE Transactions on Nuclear Science, February 1985, Vol. NS-32, page 305.
- [3] Fujitsu part number MB114F307.



Collection: ☒ Journals ☒ Conferences ☒ Standards

Your search matched 1 of 608497 documents.

1 are presented on this page, sorted by Score in descending order.

<u>DOC TYPE</u>	<u>VIEW ISSUE</u>	<u>VIEW FULL PAGE</u>	<u>VIEW CITATION</u>
	<u>TOC</u>		
CNF			<u>Recent improvements to the DIII-D neutral beam instrumentation and control system</u> <i>Kellman, D.H.; Hong, R.</i> Fusion Engineering, 1997. 17th IEEE/NPSS Symposium Volume: 2 , 1998 , Page(s): 807 -810 vol.2

| [IEL Online Home](#) | [Search](#) | [Advanced Search](#) | [What's New](#) | [Help](#) | [Logout](#) |
| [FAQ's](#) | [Support](#) | [Comments](#) |

Copyright 1999 Institute of Electrical and Electronics Engineers. All rights reserved.

SCALABLE COHERENT INTERFACE

Knut Alnæs & Ernst H. Kristiansen, Dolphin Server Technology A.S., Oslo, Norway

David B. Gustavson*, Stanford Linear Accelerator Center, Stanford, California

David V. James, Apple Computer, Cupertino, California

Abstract

The Scalable Coherent Interface (IEEE P1596) is establishing an interface standard for very high performance multiprocessors, supporting a cache-coherent-memory model scalable to systems with up to 64K nodes. This Scalable Coherent Interface (SCI) will supply a peak bandwidth per node of 1 GigaByte/second. The SCI standard should facilitate assembly of processor, memory, I/O and bus bridge cards from multiple vendors into massively parallel systems with throughput far above what is possible today.

The SCI standard encompasses two levels of interface, a physical level and a logical level. The physical level specifies electrical, mechanical and thermal characteristics of connectors and cards that meet the standard. The logical level describes the address space, data transfer protocols, cache coherence mechanisms, synchronization primitives and error recovery. In this paper we address logical level issues such as packet formats, packet transmission, transaction handshake, flow control, and cache coherence.

1 INTRODUCTION

The Scalable Coherent Interface (SCI) Project started in November 1987 as a study group under the Microprocessor Standards Committee (MSC) of the Technical Committee on Mini- and Microcomputers in the IEEE Computer Society. Paul Sweazey was the chairman of the study group, which used the working name SuperBus. In July 1988 the study group became a working group, adopting the name Scalable Coherent Interface, chaired by David B. Gustavson.

The objective of the SCI working group is to define an interconnect system which scales well as the number of attached processors increases, provides a distributed cache-coherent memory system, and defines a simple interface between modules [1,4,5,7,8,11].

We quickly discovered that a traditional backplane bus could not achieve our goals. Today's buses are limited by the distance a signal must travel and the propagation delay across a backplane. In asynchronous buses, the limit is the time needed for a handshake signal to propagate from the sender to the receiver and for a response to return to the sender. In

synchronous buses, it is the time difference between clock and data signals which originate in different places.

Transmission lines in backplanes are disturbed by connectors and variations in loading as the number of inserted modules varies. This makes reliable high speed signalling on a backplane bus very difficult. In addition, a backplane bus can only service one request at a time and therefore becomes a bottleneck in multiprocessor systems.

The SCI working group solves these problems by defining a radically different interconnect system. We are defining an interface standard which enables a system integrator to attach his board to an interconnect which may have many different configurations. These configurations may range from simple rings to complex multistage switching networks.

The interface standard defines a point-to-point communication between neighbor nodes, greatly reducing transmission line problems. This point-to-point link uses differential ECL signalling, allowing high speed transfers of 1 Gbyte/second though the link is only 2 bytes wide. Small packets carry data from node to node across these links. Buffering in the node interfaces accommodates many simultaneous requests, making SCI well suited to high performance multiprocessor systems. The SCI standard allows up to 64K nodes to be connected to an interconnect, and should provide the next generations of computers with sufficient interconnection bandwidth.

A bit-serial link is also under development, for use with fiber optic or coaxial cable links over longer distances (but at lower speeds). The bit serial version will support the same architecture and protocols as the 2-byte-wide version.

Cache coherence is an important part of the proposed standard. Current mechanisms prove insufficient when the number of processors increases dramatically. This calls for a new approach to the cache consistency problem. The SCI working group is defining a scalable distributed directory scheme where processors sharing cache lines are linked together by pointers stored in the caches.

High volume products using the SCI standard are expected to become available by the mid-1990's. Figure 1 gives a rough estimate of future volumes of board level products[2].

* Work supported by the Department of Energy, contract DE-AC03-76SP00515.

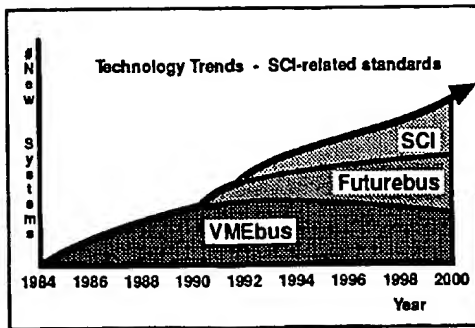


Figure 1. Technology trends.

The following sections provide more insight into the solutions which the SCI working group is currently pursuing. The next section describes various configurations of an SCI system and emphasizes interfacing via different interconnects. The packet format and packet transmission is described in section three. In section four we focus on the mechanisms for packet flow control. Section five gives a brief overview of the cache coherence model. Finally, we summarize the standardized Control and Status Register space and the status of realization in silicon.

2 CONFIGURATIONS

SCI supports multiple configurations ranging from simple low cost implementations to high performance, high cost systems. An important property of SCI is that it includes hooks to allow several different implementations to reside

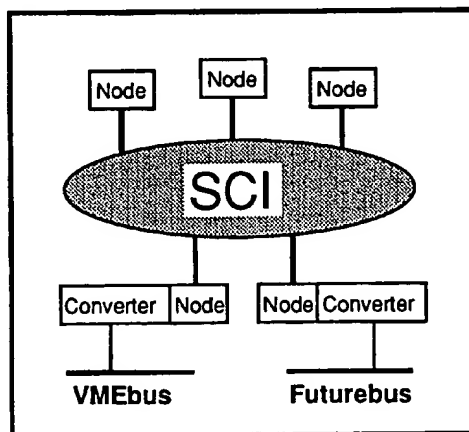


Figure 2. SCI Configuration.

simultaneously in a system. This is done by separating the interfacing node from the transporting interconnect. A view of a typical system is illustrated in Figure 2.

2.1 SCI viewed by a node

An SCI node receives a steady stream of data and transmits another stream of data. These streams consist of SCI packets and idle symbols. A node is responsible for operating on these packets and idle symbols according to the SCI standard. To do that, a node may have the construction shown in Figure 3.

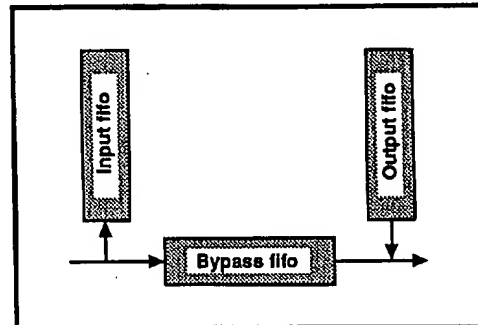


Figure 3. SCI interface.

When there is no traffic on the SCI interconnect, a node receives idle symbols. Since the utilization is zero in this case, all nodes are free to transmit. The idle symbols convey this information to the nodes. In case the node has nothing to send and the bypass fifo is empty, the output consists of idle symbols only.

When a node receives a packet, it checks the packet's destination. Packets destined for other nodes are routed to the bypass fifo and transmitted onward. The retransmitted packet accumulates flow-control information for other SCI nodes. The flow-control information is divided between the packet header and the (minimum one) idles separating the packets. The arbitration, priority and forward progress schemes are enforced this way.

When a node receives a packet which is destined for it (and it is ready to accept it), the packet is routed to the input fifo until the node has time to process it further. The packet's header information is also used to generate a short 'echo' packet, which is routed to the bypass fifo, ultimately to be received by the packet's sender. The echo is part of the arbitration, priority and forward-progress mechanisms.

A node which is granted interconnect access and which has an empty bypass fifo is allowed to transmit a packet. Since many nodes may have interconnect access simultaneously, multiple nodes may transmit at the same time. This contention is solved either by buffering in the interconnect or by filling

the bypass fifo of the transmitting node(s). The SCI system uses idles, packet headers, and echoes to selectively grant interconnect access under heavy system loading.

2.2 SCI interconnect

SCI can be configured in many ways. However, there are two basic structures—the ring and the switch. The ring implementation is the simplest. In a ring, nodes pass packets to their neighbors. In such a structure there are no active components except the nodes. This means that the nodes themselves have to control the arbitration, priority and forward progress schemes.

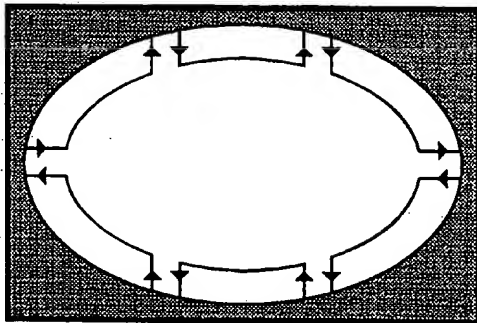


Figure 4. Ring interconnect.

A switch looks at the destination address and routes the packet directly to the destination. A switching structure can have various complexities and costs, including full crossbar switches and butterfly switches. In a switching structure, priority and forward progress schemes must be enforced by the switch. However, the node interfaces are the same in both a ring and a switch implementation.

2.3 Interconnection to other buses

Another important feature of SCI is the ability to interface to other buses. Some SCI transactions and cache states are specifically defined to accommodate other buses.

A bus bridge will respond to a range of destination addresses. The bus bridge node is responsible for converting SCI transactions into native bus transactions. Two cases are handled with special care: bus locking and cache coherence.

Most backplane buses accommodate a unique read-modify-write transaction to manipulate semaphores and other critical data. During the read transaction a lock signal is asserted, inhibiting the use of the bus until the data is written. Since SCI is defined with a four-phase transaction protocol with no guaranteed delivery order, a lock must be executed as a single SCI transaction.

Some bus protocols also incorporate a cache coherence scheme. Most use a snooping scheme where bus interfaces monitor all bus activity and update their cache states accordingly. In SCI this is not possible, since no one node can observe all the relevant transactions.

2.4 Scalability

A significant aspect of SCI is scalability. It should be possible to have a simple, cheap system with the same basic properties as a high performance one. To achieve this, a large and important task of the SCI working group is to assure that enough, but not too much, functionality is included in the standard.

A simple and cheap system would be a ring, with all packets at the same priority. This results in round-robin arbitration. A requesting node is simplified by allowing only one packet outstanding at any time, but it still needs separate request and response queues. A responding node might only be able to handle a single request at a time. If it is busy, a busy echo will inform the sender to re-transmit the packet.

A more complex, but still fairly inexpensive, system could use a combination of rings and bridges. The rings would be used between nodes which require low latency and where the ring bandwidth is sufficient. The bridges would be used to connect rings. Such a system could even support a dynamic interconnect where any node can be plugged into any socket. Multiple outstanding requests and live insertion might be supported.

The most complex system would be a switching interconnect built of elements like the butterfly switch. This interconnect is hardwired, so a node can only be plugged into its addressed location. This kind of interconnect would handle more traffic, and multiple outstanding requests from a requesting node could be supported. In addition to the round-robin arbitration scheme, multiple priority levels could be

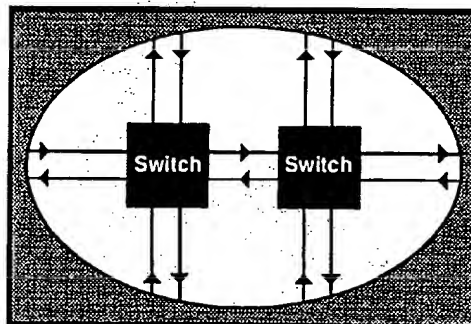


Figure 5. Switch interconnect.

provided. This interconnect also supports live insertion and withdrawal, and may be able to implement request-combining schemes to reduce the effect of congestion at hot spots.

3 PHYSICAL LAYER

SCI specifies signals at an interface to an interconnect system. All signals are unidirectional differential 100k ECL compatible signals. 18 signals are sent from a node: 16 data signals, 1 flag bit and 1 clock signal. The frequency of the clock is 250 MHz. The skew between the signals is one of the most critical items.

Power distribution is solved by distributing 48 VDC to all nodes and using on-board power converters. This reduces the number of pins needed for power and ground, allows the vendor to select the optimal voltages for various logic families and interface needs, greatly simplifies power-on module replacement, and makes uninterruptible power supplies very simple via storage batteries.

The board size recommended is 6U (233.35mm) × 280mm.

4 PACKET FORMAT

Figure 6 shows the packet format. The width of a packet word is 16 bits. In addition, a flag indicates that a packet is being received or transmitted. Each word in the packet is clocked with a differential clock line. A node receives 2 bytes at a rate of 500MHz resulting in an interconnect bandwidth of 1 Gbyte/second.

A packet consists of three main sections: a header section, an address and data section, and an error check word. The first 16-bit word of the header contains the ID code of the final receiving node. By looking at the first word of a packet, a node can quickly determine if the packet is addressed to that node. During routing through an SCI interconnect,

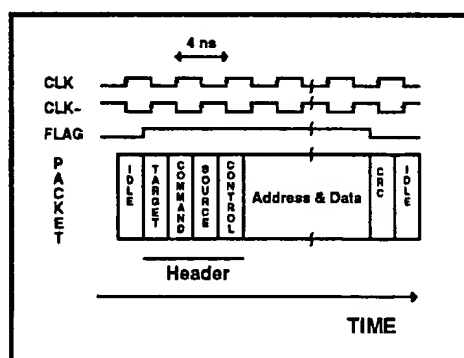


Figure 6. Packet format.

intermediate nodes and switches look at the target word to determine where to route the packet. The third word of the packet contains the ID code of the sender, needed to address the response back to the correct sender, as shown in Figure 7.

The command word of the header controls packet flow and interconnect access. Priority arbitration is supported with round robin arbitration on the lowest level. Flow control and arbitration will be discussed in more detail in section 5. The command word of the header also contains the transaction type and the packet length.

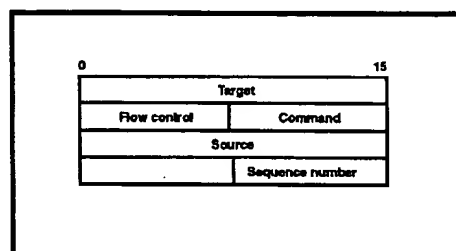


Figure 7. Header format.

The command field contains the command a responder must execute. In a multiprocessor SCI environment, a command is often applied to a cache line. The cache line size is 64 bytes, but manipulations on smaller and larger data sizes are also supported. The commands can be divided into cache coherence transactions, lock transactions, DMA transactions, and I/O register transactions. The cache coherence transactions manipulate a linked-list structure used to maintain a coherent memory image.

The sequence number in the control word is a label which identifies a packet. A node connected to an SCI interconnect may send many requests (up to 64), before a response is received. This transaction pipeline can cause responses to be returned out of order, and therefore a sequence number is needed to identify a response with the corresponding request.

The target word and the three first address words define the 64-bit SCI address. The data part may contain from 16 to 256 bytes. When a packet is transmitted, a cyclic redundancy code (CRC) for the packet is computed, and this code is attached after the last word of the packet. The CRC is a "serial-parallel" version of the 16-bit CCITT-CRC.

4.1 Packet reception

In an SCI interconnect, a node is addressed by a 16-bit identification code, which is located in the first word of the packet. This allows 64K nodes to be attached to the interconnect. This allows for easy detection, and decisions to pick up the packet can be made quickly. An input flag marks the beginning of a packet; if the target ID of the packet

matches the ID code of the node, the packet is stripped from the interconnect. While the packet is being stripped and received, a CRC for the packet is computed. The computed CRC code is compared with the CRC code at the end of the packet. If they match, the reception is completed; otherwise the packet is discarded.

A stripped packet creates a small echo packet, with interchanged target and source IDs. The echo packet is returned to the sender for flow control. If the input fifo was not empty, the busy bit in the control word of the echo is set, so that the sender knows it must retransmit the packet later. If bad CRC is received, the echo CRC is complemented so it will be discarded (it is too late to avoid its transmission).

4.2 Packet transmission

A node may transmit if the bypass fifo is empty (see Figure 3) and the node is granted interconnect access through the flow control mechanism. Before transmission, the packet is put into the output fifo.

Transmission starts by putting the target word onto the output and setting the output flag high. The output flag is high while the packet is being transmitted. A CRC is attached to the end of the packet when the output flag goes low.

If a packet enters the node interface during transmission, and the packet is not for this node, the packet is put into the bypass fifo until the transmission is done. The size of the bypass fifo must therefore be at least as large as the maximum transmitted packet size to avoid fifo overflow.

4.3 Transaction handshake

SCI supports a transaction pipeline up to 64 transactions deep. This means that a node may send up to 64 requests without waiting for a response. A normal transaction consists of two subactions, a request subaction and a response subaction. Together with each subaction there is an echo

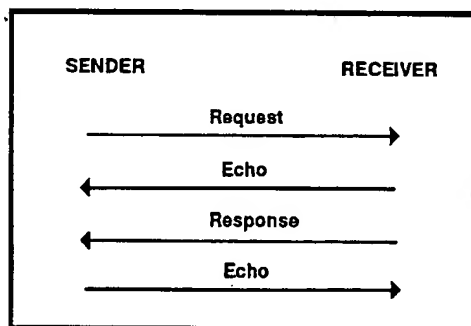


Figure 8. Split transaction handshake.

packet returned to the sender, as shown in Figure 8.

When the request is transmitted, it is labelled with a sequence number. The ID code of the sender and the sequence number uniquely identify a packet in the SCI interconnect. When a responder accepts a packet, the sequence number in the request packet is saved. The responder will add this sequence number to the response packet when the response is transmitted back to the sender.

Transmission errors could cause many kinds of problems. Fault recovery has been carefully considered, and most of the burden placed on software error handlers. The principle relied on is that transmission errors are detected by a time-out mechanism so the sender can retry a transaction if no echo or response has been received within the time-out interval.

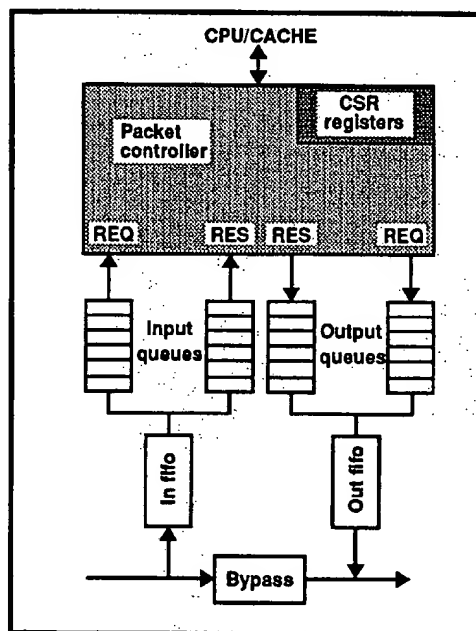


Figure 9. Node interface.

5 FLOW CONTROL

In SCI, flow control of packets is needed to maintain high throughput and fair access when many packets are sent to the interconnect at the same time. The flow control issues discussed in this section are arbitration, deadlocks, servicing, and congestion.

As explained earlier, a node may transmit when its bypass fifo is empty. This means that up to 64K nodes may start to transmit at once, allowing 64K packets to exist in the interconnect. However, nodes connected to a ring can not

retransmit until their bypass fifos are empty. To avoid starvation, an arbitration algorithm ensures that all nodes have access to the ring. Our current algorithm is based on fair and priority transactions. The arbitration mechanism is enforced by header information and idle symbols between packets. The priority level of a transaction is coded into the command word of the packet header (as shown in Figure 7).

Another node which wants to transmit and has a higher priority marks the header of a passing packet. This informs the packet's sender that another node with a higher priority wants to transmit. This flow-control information is also distributed to others, in idle symbols between the packets.

To avoid deadlocks, separate request and response queues are added to each input and output fifo as shown in Figure 9. To ensure fairness, packets are selectively accepted into these queues, based on an approximate packet aging protocol. Also, the acceptance protocol can be influenced by the incoming packet's priority.

6 CACHE COHERENCE

High performance processors need local caches to reduce the effective memory access latency. In a multiprocessor environment this leads to potential conflicts because several processors may simultaneously want to modify locally cached copies of the same data.

Cache coherence protocols define mechanisms that guarantee consistent data even if data is cached and modified by several processors. The SCI definition supports a hardware-based cache coherence protocol, reducing the programmer's software effort to secure consistency, and also reducing operating system complexity.

Many existing cache coherence protocols use a snooping technique and rely on transactions like broadcast and eavesdropping to guarantee data consistency. In a large high speed distributed system, the broadcast transaction is ineffective at best, and eavesdropping is impossible to implement because it requires a bus common to all processors in the system. Since a highly scalable interconnect system is one of the main objectives in defining the SCI, these and similar mechanisms are unsuitable.

We have developed a directory-based cache coherence protocol[6] with distributed properties, where all the nodes with cached copies participate in the control. The principle is that every sharable block in memory is associated with a list of processors sharing that block. A memory block is usually the size of a cache line, which is 64 bytes.

The selection of 64 bytes as the cache line size is based on many factors. The density of current state of the art ECL chips prohibits packet sizes larger than 80 bytes because of the fifo buffering. An 80-byte maximum packet size has a reasonable overhead, making cache line transfers efficient for a 64-byte

line size and less efficient for a 32-byte line size. Concern about false sharing makes a 128-byte line size less attractive, and trace driven simulations [10] show that a 64-byte line size is a good choice for SCI. Futurebus+ has also selected 64 bytes, making the interface between SCI and Futurebus+ simpler and more efficient.

Every block has a tag which includes a pointer to the processor at the head of the list. Each processor cache tag has a pointer to the next node sharing that cache line. In effect, all nodes with cached copies of a memory block are linked together by these pointers. The nodes have a forward pointer and a backward pointer to connect them with the previous and next node in the list. The resulting doubly linked list is shown in Figure 10.

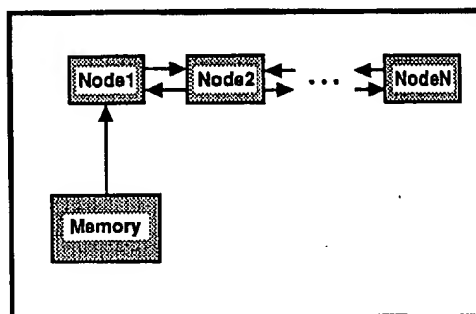


Figure 10. SCI sharing list.

This distributed list concept ensures good scaling properties. Even as the number of nodes in a list grows dramatically, the corresponding memory tag size is constant. However, two pointer locations are associated with every cached block in a node.

The list pointers are actually the interconnect addresses for the processors. When a node accesses memory to get a copy of shared data, it provides memory with its own address. If there are currently no nodes with cached copies, the requesting node is made the head of a new list and memory saves the node address in the tag for this block. If, however, there exist nodes with cached copies of data, the pointer to the head of the sharing list is returned from memory to the requesting node, and this node inserts itself at the head of the list. Currently cached data is always returned from the old head, rather than from memory.

The nodes in a linked list typically have read access to shared data. When a node wants write access, and it is currently the list head, then it purges the rest of the list. If it is in another portion of the list, the node first deletes itself from the list, then performs another memory read to move to the

head of the list. Write access is restricted to the head node only.

All bus transactions concerning cache coherence are implemented within the standard packet format described above.

The cache coherence protocol described above has not yet been tried in real systems. We are therefore relying on several people at the University of Oslo who are using their expertise to do formal verification[9].

7 CONTROL AND STATUS REGISTERS

The Control and Status Registers (CSRs) are an important part of the proposed standard. The CSR definitions are essential for all initialization and exception handling. Some of the CSRs must be SCI specific, but the majority of the necessary definitions can be common with other standards[3]. The IEEE has approved a request for a standard project for defining CSRs. The project number is IEEE P1212, chaired by David V. James. The CSR standard is being coordinated with Futurebus+, Serial Bus and SCI. It will also try to coordinate with the ongoing CSR activity for VMEbus.

8 REALIZATION

Realization in commercial systems is important for acceptance of a defined standard. Therefore the first implementation is being done in parallel with the standardization work. So far we have done measurements that assure us that it will be possible to make implementations for the 1 Gigabyte/second transfer rate.

We have both a high level and a low level simulation model of an SCI system running. We have simulated both the arbitration and the cache coherence scheme. The length of a maximum data packet will initially be limited to 64 bytes (i.e. a cache line). For the first implementation we are using ECL gate arrays, with one chip (or perhaps two) for the SCI interface and the cache coherence protocol. This interface chip will be common for all nodes. In addition, Dolphin Server Technology is making a physically addressed cache controller which can be used as a second level cache controller, and a global memory controller chip that supports the necessary directory handling in global memory.

The first configuration will be a ring structure with high performance CPU's, large main memory and connection to standard buses like VMEbus for I/O functions. We expect to have prototypes ready for testing late this year.

9 CONCLUSION

This paper has presented an overview of the objectives of the SCI working group, and the solutions which are currently being pursued. Scalability of a system is a key aspect as many high performance computer manufacturers are moving toward large multiprocessor systems. In order to utilize these systems

efficiently, a cache coherence mechanism must have good scaling properties. Also, for a system to both be cost effective and support high performance solutions, it is necessary to separate the module interface from the interconnect implementation.

We feel that our current proposals meet these objectives. The SCI project is moving rapidly and has attracted participants from many of the high performance computer companies. We already have a first draft of the standard available, and we hope to send it out for ballot late this year. The proposed architecture appears to be achievable based on technology available today.

If you would like to participate in this work, or if you would like more detailed information, please contact one of the authors or the chairman of the project:

David B. Gustavson, IEEE P1596 Chairman
Computation Research Group, bin 88
Stanford Linear Accelerator Center
Stanford, CA 94309, USA
tel: (415) 926-2863
fax: (415) 961-3530 or (415) 926-3329
Email: DBG@SLACVM.bitnet

10 ACKNOWLEDGMENTS

Many people have already contributed to SCI's development; though we cannot list them all, we wish to acknowledge a few contributions which seem to us to be particularly significant.

Paul Sweazey, originally of National Semiconductor and recently of Apple Computer, started the SuperBus study group, which he chaired until the SCI working group was organized. Paul has also brought a thorough understanding of the cache coherence problem, due to his work coordinating the Futurebus Cache Coherence task group.

Paul Borrill of Sun Microsystems, Futurebus+ chairman, helped push our goals to much higher bandwidths and to increased parallelism through the use of switches instead of shared buses.

John Moussouris, a co-founder of MIPS Computers, has provided critical insights into the directions we need to take in order to rendezvous with future technology, has helped put us in touch with the appropriate experts, and has helped expose problems and errors in various models.

Phil Ponting of CERN in Geneva has provided effective and vital communications and redistribution to the many European participants.

Hans Wiggers of Hewlett Packard Laboratories has helped us examine various physical layers, and is considering the implications of an optical fiber implementation of SCI.

Mark Williams of Hewlett Packard leads a joint task group to consider the interface between SCI and Futurebus+.

Stein Gjessing, Stein Krogdahl and Ellen Munthe-Kaas at the University of Oslo are working on formal specification and verification of the SCI cache coherence protocols.

11 AUTHORS

Knut Alnæs, originally of Norsk Data and now a staff member of Norsk Data subsidiary Dolphin Server Technology A.S., Oslo, Norway, is involved in SCI simulation and is responsible for the interface chip design.

David B. Gustavson works in the Computation Research Group of the Stanford Linear Accelerator Center at Stanford University, Palo Alto, California. He is chairman of the IEEE P1596 (SCI) working group. He has had experience with many standard buses, beginning with the early S-100 (IEEE 696) but especially Futurebus (IEEE P896.x) and Fastbus (IEEE 960-1986, IEC 935). He also chairs the Fastbus Software Working Group (IEEE 1177-1990).

David V. James, originally of Hewlett Packard and recently of Apple Computer, Cupertino, California, has broad experience with multiprocessor architecture, which he is applying to SCI's needs, from control register and I/O architecture to distributed cache coherence and forward progress. David is vice chairman of SCI, coordinator of the Logical task group and has written the majority of the working documents. He is also the chairman of the CSR project (IEEE P1212).

Ernst Kristiansen was responsible for development of memory systems, proprietary buses and I/O-systems at Norsk Data, and is now at Norsk Data's subsidiary, Dolphin Server Technology A.S., Oslo, Norway, where he is project manager for the SCI implementation.

12 REFERENCES

1. D. B. Gustavson, D. V. James, J. Moussouris and P. Sweazey, "The Scalable Coherent Interface Project (SuperBus)", draft of August 22, 1988.
2. P. L. Borrill "VMEbus—The Next 5 Years", VMEbus in Research, October 1988.
3. D. V. James, "Scalable I/O Architecture for Buses", COMPCON Spring 1989, pp 539-544.
4. D. B. Gustavson, "Scalable Coherent Interface", COMPCON Spring 1989, pp 536-538.
5. E. H. Kristiansen, "IEEE SCI (P1596)", VMEbus in Research, October 1988.
6. A. Agarwal, R. Simoni, J. Hennessey and M. Horowitz, "An Evaluation of Directory Schemes for Cache Coherence", 15th International Symposium on Computer Architecture, June 1988.
7. E. H. Kristiansen, K. Alnes, B. O. Bakka and M. Jenssen, "Scalable Coherent Interface", Eurobus Munich, May 1989.
8. P. Sweazey, "Cache Coherence on SCI", IEEE/ACM Computer Architecture Workshop, Eilat, Israel, June 1989.
9. S. Gjessing, S. Krogdahl, E. Munthe-Kaas, "Formal Specification and Verification of SCI Cache Coherence", NIK 89, Stavanger, Norway, November 1989.
10. H. O. Bugge, E. H. Kristiansen, "Trace Driven Simulations for Decisionmaking on Cache Line Size and Cache Size in a Two Level Cache Design", NIK 89, Stavanger, Norway, November 1989.
11. D. B. Gustavson, "The Scalable Coherent Interface, IEEE P1596, Status and Possible Applications to Data Acquisition and Physics", IEEE Nuclear Science Symposium, 1989.